System, Method, and Program Product for Extracting a Multiresolution Quadrilateral-Based Subdivision Surface Representation from an Arbitrary Two-Manifold Polygon Mesh

FIELD OF THE INVENTION

This invention relates to the field of computer graphics. More specifically, the invention relates to converting arbitrary polygon meshes to multiresolution subdivision surfaces.

DEFINITIONS

The following formally define fundamental mathematical concepts used throughout this description. See any standard textbook on fundamental concepts of mathematics and computational geometry.

An n-*manifold* is a topological space which is locally Euclidean, i.e., around every point, there is a neighborhood which is topologically the same as the open unit ball in $\Re^n$, $n \geq 2$.

5

A 2-manifold is an n-manifold with n = 2.

A map $\phi : A \rightarrow B$ is said to constitute a *homeomorphism* between A and B if it is continuous, bijective, and has a continuous inverse.

10   A *coordinate chart* is a way of expressing the points of a small neighborhood on a manifold M as coordinates in Euclidean space. Formally, a coordinate chart is a map: $\phi : U \rightarrow V$ where U is an open set in M, V is an open set in $\Re^n$ and n is the dimension of the manifold. The map ϕ must be a homeomorphism.

15   *An atlas* is a collection of consistent coordinate charts on a manifold. As the name suggests, an atlas corresponds to a collection of maps, each of which shows a piece of a manifold and looks like the flat two-dimensional Euclidean space. To use an atlas, one needs to know how the maps overlap. To be useful, the maps must not be too different on these overlapping areas.

A *Voronoi diagram* (or *Dirichlet tessellation*) is a partitioning of an input data set around n points called *seeds* into n regions (also known as *cells*) such that each region contains exactly one point and every point in a given region is closer to its seed than to any other point. A *centroidal Voronoi diagram* (CVD) has the property that each seed lies at the center of its region (cell). *A constrained centroidal Voronoi diagram* (CCVD) has the property that the seed points are the constrained mass centroids of their associated regions (see also Du et al., Constrained Centroidal Voronoi Tessellations for Surfaces, SIAM Journal for Scientific Computing, 24(5), pp. 1488-1506, 2003).

A *scalar field* is a continuous function $\Phi : \Omega \to \Re$, where Ω is a connected domain in $\Re^n$, $n \geq 1$. The image of Φ embedded in $\Re^{n+1}$ space, i.e., $\Im = \{(X, \Phi(X)) \mid X \in \Omega\} \subset \Re^{n+1}$ is called a *hypersurface*. When $n = 2$, $\Im$ is called a *height field*.

The *medial axis transform* (MAT) of an object is defined as the set of maximal balls completely contained within the object. The *medial axis* (MA) consists of the centers of the balls and intuitively can be viewed as the skeleton of the object.

Subdivision defines a smooth surface recursively as the limit of a sequence of meshes. Each finer mesh is obtained from a coarser mesh by using a set of fixed refinement rules. Examples of suitable rules are the Catmull-Clark

5 subdivision rules (see E. Catmull and J. Clark, Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes, CAD 10(6), pp. 350-355, 1978). In the preferred implementation Catmull-Clark rules are used, but this is not a limitation upon the practice of this invention. Any quadrilateral-based subdivision rules for 3D models can be used.

10

Multiresolution subdivision surfaces extend subdivision surfaces by introducing details at each level. The coarsest-level mesh is sometimes also called the base mesh. Each time a finer mesh is computed, it is obtained by adding detail offsets to the subdivided coarser mesh. If given a semi-regular

15 mesh, i.e., a mesh with subdivision connectivity, one can easily convert it to a multiresolution surface by defining a smoothing operation to compute vertices on a coarser level from a finer level. The details are computed as differences between levels.

The *valence* of a vertex of a mesh is defined as the number of mesh faces adjacent to that vertex. An interior vertex of the base mesh is called an *extraordinary vertex* if it has valence different than four. All other interior vertices in a semi-regular representation have valence four and are called

5   *regular* vertices. Vertices on the boundary of an open mesh are considered regular if they have valence two and extraordinary otherwise. A mesh with subdivision connectivity is also called a *semi-regular* mesh.

10   BACKGROUND OF THE INVENTION

Depending on the application, mesh decompositions may have different requirements regarding the properties of the constituent elements. For

15   example, for skeleton extraction, prior art methods produce decompositions with complex components, possibly with higher than zero-genus. [See S. Katz and A. Tal, ACM TOG. Special issue for SIGGRAPH conference, Proceedings of Siggraph 2003, 2003]. Such decompositions may be suitable for restricted problems such as approximate skeleton extraction, but would

20   require further, non-trivial processing for most other applications. For

example, texture mapping involves associating 2D color images with regions of a 3D mesh. To establish such an association, an atlas consisting of charts that can be easily mapped to the planar domains of the images is necessary. The same holds true for semi-regular remeshing problems, where the charts

5   serve as faces of a coarse polyhedral base domain over which the remeshing is performed.

Attention is drawn to the problem of finding atlas decompositions which satisfy several important properties:

10

1. The number of patches in the atlas is small. Typically, the complexity of algorithms processing the patches is proportional to the number of patches. Obviously, for efficiency reasons, it is highly desirable to keep this number as small as possible.

15

2. Patches should be compact, with nice and simple shapes. For instance, it is common practice for textures to be packed into a single, nearly square image for rendering purposes. Shapes with irregular boundaries make packing inefficient and waste texture memory. Also, it becomes easier to apply editing operations to textures directly when

20

they appear as humanly recognizable pieces. Similarly, for the

purpose of remeshing it the faces of the base domain faces should be

nicely-shaped polygons. Last but not least, for parameterization

purposes, a simple shape often translates into a simpler

5   parameterization algorithm and reduced distortion.

Prior methods for mesh decomposition can be classified into two main

categories, according to whether they require some degree of user

involvement in defining the component charts or not. Semi-automatic

10   methods require manual outlining of one or more chart boundaries or some

other defining properties of the charts. (See for example V. Krishnamurthy

and M. Levoy, Fitting Smooth Surfaces to Dense Polygon Meshes,

Proceedings of SIGGRAPH 96, pp. 313-324, 1996.) Fully automatic

methods produce atlases without the need for interactive adjustment of

15   charts. Our proposed method falls into the latter category and solves a

significant number of problems which prior art fails to address.

20

PROBLEMS WITH THE PRIOR ART

Many atlas decomposition methods produce irregularly shaped charts which

5   are not suitable for remeshing and may be used only with difficulty, loss in

efficiency and / or additional processing for texture mapping. Most such

methods take a greedy approach and generate charts by normal bucketing,

accumulating triangles of the input mesh one-by-one while checking the

orientation of the surface normal. Such approaches can result in a large

10   number of patches for objects with many small features. In most cases, there

is not enough control over the shape of the charts, which can become rather

irregular. As a result, they cannot be used directly for mesh extraction. See

M. Garland, A. Willmott, and P. Heckbert, Hierarchical Face Clustering on

Polygonal Surfaces, Proceedings of ACM Symposium on Interactive 3D

15   Graphics, 2001.

The majority of existing automatic methods for semi-regular remeshing

target triangle-based subdivision schemes and are not applicable to

quadrilateral remeshing. They are typically based on a mesh simplification

20   scheme which takes a fine triangle mesh and decimates it into a coarse

triangle mesh. The coarse triangle mesh is then used as a base mesh for Loop

subdivision. (See, for example, A. W. F. Lee, W. Sweldens, P. Schroeder, L.

Cowsar, and D. Dobkin, MAPS: Multiresolution Adaptive Parameterization

of Surfaces, Proceedings of SIGGRAPH 98, pp.95-104, 1998 and T. Kanai,

5   "MeshToSS: Converting Subdivision Surfaces from Dense Meshes", Proc.

Modeling and Visualization 2001", pp. 325-332, 2001). There is no

straightforward process that would adapt this process to quadrilateral

remeshing as there is no simplification scheme that decimates a polygon

mesh into a coarse quadrilateral mesh.

10

The human eye is sensitive to the variations of light intensity reflected from

an object.

The light intensity depends on the angle between the object surface normal

and a light source. Shape variations that result in large variation variations

15   in the surface normal – changes in curvature such as ridges, creases or

ravines in the surface -- are then particularly noticeable to a human

observer, that is they are *perceptually salient*. Existing methods typically

produce charts that are not specifically aligned with the salient features of

the input model. A large number of charts is hence required to ensure a good

approximation after remeshing with a reasonable number of subdivision levels.

Some methods have been proposed that produce a quadrilateral base mesh by first generating a triangle base mesh followed by a grouping of triangles into quadrilaterals. (See M. Eck and H. Hoppe, Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological type, Proceedings of Siggraph 1996, pp. 325-334, 1996). In addition to being inefficient, such an approach is not always guaranteed to work: the triangle mesh would have to have an even number of triangles and even then, there is no guarantee that a complete pair wise matching of triangles into quadrilaterals can be found. No alignment to salient model features is sought.

Finally, for texture mapping, if the charts are not height fields, they first have to be flattened before the texture can be applied. Hence, the texture images corresponding to the charts are distorted by flattening and they do not appear the same as they do on the model. This makes it difficult to do touch up edits of the textures in 2D, having to work instead with the texture applied to the 3D model.

All above cited references are herein incorporated by reference in their

entirety.

ASPECTS OF THE INVENTION

5

An aspect of this invention is an improved system and method for

converting arbitrary polygon meshes to multiresolution subdivision surfaces.

10 An aspect of this invention is an improved system and method for

converting arbitrary polygon meshes to multiresolution subdivision surfaces

accounting for model features.

An aspect of this invention is an improved system and method for

15 converting arbitrary polygon meshes to quadrilateral-based multiresolution

subdivision surfaces.

An aspect of this invention is an improved system and method for providing

atlases that can be used for various applications (e.g., for low distortion

20 texture mapping, for model parameterization)

SUMMARY OF THE INVENTION

5   The present invention is an improved computer system, method, and
program product that has one or more input devices for receiving one or
more input meshes representing a three dimensional model. The three
dimensional model is capable of being represented as a 2-manifold polygon
mesh. A conversion process automatically converts the input mesh to a
10   multiresolution quadrilateral-based subdivision surface (MQSS)
representation.

BRIEF DESCRIPTION OF THE FIGURES

15

The foregoing and other objects, aspects, and advantages will be better
understood from the following non limiting detailed description of preferred
embodiments of the invention with reference to the drawings that include the
following:

20

Figure 1 is a block diagram on one preferred embodiment of the present system.

Figure 2, comprises Figures 2A through 2O, and illustrates various steps in a process of conversion from an arbitrary 2-manifold mesh to a MQSS representation.

Figures 2A through 2I, illustrate an example triangle mesh being decomposed by an atlas generator process.

Figure 2J through 2K illustrate a polygonal mesh extracted from the decomposed model in Figure 2I by a mesh extraction process.

Figure 2L is a quadrilateral base mesh derived from the polygonal mesh using the mesh extraction process.

Figures 2M through 2O illustrate a multiresolution subdivision hierarchy built on the quadrilateral base mesh of 2L as a result of resampling and multiresolution analysis by a resampling process.

Figure 3 is flow chart of a conversion process.

Figure 4 is a flow chart of an atlas generator process.

5 Figure 4A is a flow chart of an example automatic seed selection process.

Figure 4B is a flow chart of an initial partitioning by means of a constrained centroidal Voronoi diagram (CCVD) computation.

10 Figure 4C is a flow chart of a refined partitioning obtained by splitting regions of the CCVD decomposition into height-field regions.

Figure 4D is a flow chart of a further refined partition obtained by splitting height field regions into approximately convex regions with disk topology.

15 Figure 4E is a flow chart of a color quantization process for classifying faces of the input mesh into classes according to their color.

Figure 5 is a flow chart of the quadrilateral base mesh extraction process.

20

Figure 6 is a flow chart of the resampling and analysis process.

Figure 7 further illustrates the preferred seed selection process described in Figure 4A.

5

DETAILED DESCRIPTION OF THE INVENTION

This invention relates to the remeshing of three dimensional models
10   represented by 2-manifold polygon meshes. The remeshing method is automatic and efficient, allowing the processing of large meshes.

Figure 1 is one example and simplified block diagram of a preferred data processing system 100 that is suitable for practicing this invention. The data
15   processing system 100 includes at least one data processor 101 coupled to a bus 102 through which the data processor 101 may address a memory sub-system 103, also referred to herein simply as the memory 103. The memory 103 may include RAM, ROM and fixed and removable disks and/or tape. The memory 103 is assumed to store a program containing program
20   instructions for causing the data processor 101 to execute methods in

accordance with the teachings of this invention. Also stored in the memory

103 can be at least one database 104 of digital 3D models, such as models of

objects, certain non-limiting examples of which are shown in Figs. 2. The

digital model data may include models obtained by scanning real objects

5   using a 3D scanner, digital models created using a computer, models

imported from existing databases, and/or models generated by or derived

from any suitable means. In general, the digital model data may be any

desired type or types of 3D models, including but not limited to digital

models of persons, places, animals, plants, mechanical parts and assemblies,

10   packaging, containers, abstract forms and so forth.

The data processor 101 is also coupled through the bus 102 to a user

interface, preferably a graphical user interface (GUI) 105 that includes any

one or more of the following well known user interface devices: a keyboard,

15   a mouse, a trackball, a voice recognition interface, and any general purpose

user display device, such as a high resolution graphical CRT display

terminal, a LCD display terminal, or any suitable other display device.

The data processor 101 may also be coupled through the bus 102 to a

20   network interface 106 that provides bidirectional access to a data

communications network 107, such as an intranet and/or the internet.

Coupled to the network 107 can be one or more sources and/or repositories

of digital models, such as a remote digital model database 108 that is

reachable through an associated server 109.

5

The data processor 101 is also preferably coupled through the bus 102 to at

least one peripheral device 110, such as a scanner 110A (e.g., a 3D scanner)

and/or a printer 110B and/or a 3D model making apparatus, such as a rapid

prototyping system, and/or a computer controlled fabrication system.

10

In general, this invention may be implemented using one or more software

programs running on a personal computer, a server, a microcomputer, a

mainframe computer, a portable computer, and embedded computer, or by

any suitable type of programmable data processor 101.

15

Given this disclosure, many other system architectures would become

known to one skilled in the art.

The use of this invention substantially improves the remeshing of 3D model

20   data for many applications (for example, model editing and compression).

The teachings of this invention can also be configured to provide high-quality atlas decompositions of 3D models for various applications (for example, texture mapping).The methods may be used to process the digital 3D model data stored in the 3D model database 104a and/or in the remotely

5   stored 3D model database 108 over the network 107 and in cooperation with the server 109. As but one example, a 3D input model to be remeshed could be remotely stored in the 3D model database 108a, while the remeshed 3D model represented as a multiresolution subdivision surface could be stored in the local 3D model database 104.

10

In a preferred implementation, the atlas 104b, 108b may be implemented as a collection of clusters of faces of the input mesh, each cluster corresponding to a chart of the atlas and containing pointers to mesh faces that are part of it. Typically, this data would be stored in memory 103 as shown in Figure 1.

15   However, any other data structures representing the atlas are possible and are covered by the present invention along with other methods of storing them, possibly at a remote location 109 on a network.

The invention includes a conversion process 300, shown in more detail in

20   Figure 3.

Figure 2 is a series of example model illustrations showing the results of performing the steps of the conversion process 300. The parts of Figure 2 are: (A) an example input model, (B) a wireframe rendering illustrating the

5   underlying arbitrary triangle mesh representation of the input model, (C) face normals encoded as colors per face over the model, (D) smoothing of colors to remove noise and tessellation artifacts, (E) initial decomposition by color quantization, (F) cleanup of previous partition (notice some straightening of chart boundaries and removal of small charts), (G)

10   centroidal Voronoi diagram around the centers marked with black dots in (F) (note that somecharts are not height fields), (H) splitting of charts into height fields, (I) additional cleanup and boundary straightening, (J) boundaries of charts are extracted to form a base domain for use in remeshing with subdivision connectivity, (K) further cleanup to improve the coarse mesh, (L)

15   quadrilateral mesh obtained by applying one step of catmull-Clark subdivision to the mesh in (K), and (M)-(O) three consecutive levels of the multiresolution hierarchy built on top of the base domain shown in (L).

Figure 3 is flow chart of the conversion process 300 comprising the steps of

20   atlas generation 400 (described in detail in Figure 4), quadrilateral mesh

extraction 500 (described in detail in Figure 5), and resampling and

multiresolution analysis 600 (described in detail in Figure 6).

Figure 4 is a flow chart of an atlas generator process 400. Generally, a 2-

manifold polygon mesh M is input to the process 400 and the process 400

outputs an atlas A of charts partitioning M.

In process 400, an atlas decomposition is found for the input mesh using a

combination of clustering techniques as described below. The resulting

altas is made of charts that have several properties:

1. The charts are non-overlapping, i.e., forming a partition of the input

   mesh.

2. The charts correspond to simple, nicely-shaped regions of the input

   mesh.

3. Each chart approximates a height field over some planar domain.

4. Each chart is homeomorphic to a disk.

5. The network of curves delineating the boundaries of the charts forms

   a wireframe mesh in which the majority of vertices have low valence.

The process 400 comprises the following steps (with more detail given below):

1. determining an initial number of chart seeds with their locations 410, see Figure 4A, below.

2. computing a constrained centroidal Voronoi decomposition (CCVD) 420, see Figure 4B, below.

3. splitting the CCVD into height-field charts and generating a new partition 430, see Figure 4C, below.

4. splitting each region of P into approximately convex regions homeomorphic to disks and generating a new partition Q 440, see Figure 4D, below.

5. cleanup (optional) 450

Reference can also be made to Figures 2A through 2I.

To generate an atlas over the input mesh, a number N of initial seed points is generated over the mesh according to some distribution. For example, for a given N, the seed points could be generated using a random placement process that uniformly distributes points over the mesh (see R. Osada, T.

5   Funkhouser, B. Chazelle, and D. Dobkin, Shape Distributions, ACM Transactions on Graphics, 21(4), October, 2002, for an example of such process). Alternatively, a suitable number N and initial locations of the N seeds on the mesh could be estimated using a seed generation process 410 as shown in Figure 4A (see also Figure 2F and Figure 7).

10   Figure 4A is a flow chart of an example automatic seed selection process.

First, the mesh is split (412) into relatively flat regions based on information provided by the mesh normals. Mesh normals may be computed per vertex or per face. Without restricting the generality of our approach, in what

15   follows we use the term normal to mean face normal. We encode each normal vector n as a color by assigning each of the three components of the vector $n = (nx, ny, nz)^T$ to each of the red, green, and blue components of a color. If we want to visualize the normal variation over the input mesh, we can render the model with colors, as illustrated in Figure 2C. Following this

20   step, a smoothing of the colors may optionally be applied (413) to reduce

any noise present in the normal data. An example of Laplace-type smoothing of colors is shown in Figure 2D. Partitioning the model into relatively flat regions translates into identifying regions of relatively constant color. Many methods for color quantization in images exist. Some of these are suitable

5   for applying to meshes. For example, we use a MacQueen-type method (see Q. Du, M. D. Gunzburger, and L. Ju, Constrained Centroidal Voronoi Tessellations for Surfaces, SIAM Journal for Scientific Computing, 24(5), pp. 1488-1506, 2003) to perform color quantization directly on the input mesh (415). Other methods could also be used. The result of this

10   quantization step is shown in Figure 2E in which the differently colored areas correspond to regions obtained with this process. An optional cleanup step in which the boundaries between regions are straightened may be used (417). As seen in this figures, the regions generated at this step are relatively flat (by construction), but cannot typically be used for remeshing due to their

15   irregular shape. Instead we use these regions to place seed points as follows. For each region, we compute an approximate medial axis (MA) and we sample it at several positions (419). These sample points are included into the set of seeds. While the MA approximation and sampling can be performed in many different ways, we opted for a simple and efficient

20   method in which we first sample the boundary of the region and then we

iteratively generate a set of points located at the centers of region faces with

the property that their minimum distance to the boundary samples and to

maximal balls around previously generated points is largest. The number of

points thus generated depends on the size of the region. The seeds $S_1, ..., S_N$

5   are selected from this point set by sorting the points in the set according to

some criterion and imposing a threshold value with respect to which points

are accepted or eliminated. For instance, points in the point set could be

sorted in descending order according to their maximum minimum distance

to the boundary and / or other points in the set and a threshold could be

10   chosen at the point of largest decrease in distance. Another example would

be to always keep a fixed number of points from the sorted set. This process

is illustrated in Figure 7. Figure 2F shows an example set of seed points on

the input mesh of Figure 2A.

15   Figure 4B is a flow chart of an initial partitioning by means of a constrained

centroidal Voronoi diagram (CCVD) computation (430).

Having a set of N seeds on the input mesh, we compute a constrained

centroidal Voronoi decomposition (CCVD) (420) of the mesh around the

20   given seeds $S_1, ..., S_N$ (see Figure 4B). For this we use again a MacQueen-

type clustering algorithm (426) (see Q. Du, M. D. Gunzburger, and L. Ju, Constrained Centroidal Voronoi Tessellations for Surfaces, SIAM Journal for Scientific Computing, 24(5), pp. 1488-1506, 2003). Any clustering algorithm that produces a CCVD cold be used. Faces of the input mesh are

5   classified based on their distance to the seeds. A face is assigned to a particular region if and only if the distance from that face to the seed of that region is shorter than the distance from that face to the other seeds. After all faces are classified, the position of each seed is recomputed, as the projection of the centroid of the faces in its region onto the input mesh. For

10   distance computation (422) we use approximate geodesic distances over the mesh (see, for example, K. Polthier and M. Schmies, Straightest Geodesics on Polyhedral Surfaces, Mathematical Visualization, Ed: H.C. Hege, K. Polthier Springer Verlag, 1998, pp. 391, Preprint: TU-Berlin, SFB 288, Preprint No. 327, 1998). However, other types of distances may also be used.

15   For example, Euclidean distances can be used for more efficient processing, especially when the seeds are placed in the center of relatively flat regions in which case Euclidean distances approximate well geodesic ones. An example of such a decomposition is shown in Figure 2G. The cells of the constrained centroidal Voronoi decomposition thus computed form a

20   partition of the input mesh. However, the computation of the Voronoi

decomposition does not ensure that the resulting cells are height fields. For example, the mid regions on the ears of the 3D model of Figure 2 shown in Figure 2G completely surround the ear in a cylindrical fashion and thus are not height fields. Since we would like our final charts to have this property,

5   we must consider the cells one at a time and further process those that are not height fields (430). See Figure 4C.

Figure 4C is a flow chart of a refined partitioning P obtained by splitting regions of the CCVD decomposition into height-field regions (430).

10

For each region that exhibits a normal variation greater than a certain threshold (432), we further split it into height fields using a color quantization process (434) similar to the one used to produce the initial quantization, but this time we restrict it within the cell. Otherwise the region

15   remains unchanged (436). The set of all resulting regions forms the partition P (438). See also Figure 2H and note the ear regions that are now split into several regions that roughly conform with the sides of the ears (the boundaries of these regions are approximately aligned with the ear features).

Figure 4D is a flow chart of a further refined partition Q obtained by splitting height field regions of partition P into approximately convex regions homeomorphic to disks.

5   After a partition of the input mesh into height fields is obtained, a step of further ensuring that the resulting regions are nicely shaped consists of checking the shape of each region against a shape measure that quantifies whether or not the region is hoemomorphic to a disk (i.e., region boundary has exactly one connected component) and whether or not it is

10   approximately convex (440). Regions that do not satisfy a shape compactness criterion (442) are further split into compact regions homeomorphic to disks (444) using a CCVD decomposition around medial axis samples. The CCVD decomposition process 420 is applied restrictively to each region that has to be split. The seeds used in this case can be selected

15   randomly or using samples from the approximate medial axis of the region, similar to process 419 in Figure 4A. Otherwise the regions remain unchanged (446).

An optional cleanup of the charts may optionally follow (450), see Figure 4,

20   in which boundaries between charts are straightened. For example, the

straightening may be performed locally in the neighborhood of the original

boundary, by applying a constrained Dijkstra algorithm (see T. H. Corman,

C. E. Leiserson, and R. L. Rivest, Introduction to Algorithms, pp. 527,

McGrawHill, 1990) to find the shortest path between the endpoints of the

5   path within a certain distance from the original path.

The atlas thus obtained can be used directly for applications like texture

mapping and mesh parameterization. In the following we describe the

necessary processing to use the atlas decomposition for remeshing with

10   semi-regular connectivity.

It is observed that each chart of the atlas has, on average, at most six

neighboring charts. This follows from the process of constructing the atlas

using Voronoi decompositions and from the property of Voronoi diagrams

15   that the average number of Voronoi edges per Voronoi polygon does not

exceed six (see A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, Spatial

Tessellations: Concepts and Applications of Voronoi Diagrams, pp. 65,

Wiley & Sons, Ltd., 2000).

Figure 4E is a flow chart of a color quantization process for classifying faces of the input mesh M into classes according to their color forming a partition R of the input mesh M (415).

5   A set of representative colors (415a) is defined to correspond to a small subset of all possible normal directions. For example, by choosing the subset of normal directions to be the 14 normal vectors corresponding to the faces and vertices of the unit cube and by mapping these normals to colors as previously described, we obtain a set of 14 colors that can be viewed as a

10   seed set for the quantization process. All faces of the input mesh can be classified based on their color using MacQueen's approach (415b) to belong to one of the classes defined by the seeds (the one with the closest seed in color space). Note that the choice of 14 representative colors is just an example, any other number and choice of subset of normal directions may be

15   used. The seeds themselves are adjusted during the process to the centroids of their regions (this is, in effect, a CCVD in color space).

Figure 5 is a flow chart of the quadrilateral base mesh extraction process. In a preferred embodiment of process 500, the atlas decomposition A obtained

20   through process 400 is used to extract a coarse quadrilateral mesh MQ

roughly approximating the input mesh M. Reference can also be made to Figures 2J through 2L.

Having found an atlas decomposition over the input mesh with charts corresponding to approximately flat regions of the models and homeomorphic to disks, a quadrilateral base mesh corresponding to this decomposition is extracted using a quadrilateral mesh extraction process 500 as shown in Figure 5. First a set of boundary curves between the charts of the atlas are extracted (510). The curves are subsequently simplified into polylines (520). Trivial simplification to the line segment between the first and last point of each curve may typically be sufficient, due to the simple shape of the charts ensured during the atlas generation process 400. Alternatively, the simplification can be done using any suitable curve simplification method (for an example using a 3D version of the well known Douglas-Peucker algorithm; see D. H. Douglas and T. K. Peucker, Algorithms for the Reduction of the Number of Points required to Represent a Digitized Line or Its Caricature, The Canadian Cartographer, 10(2), pp. 112-122, 1973). The simplified curves bordering each chart form a polygon approximating that chart. The set of all such polygons form a coarse polygonal mesh that approximates the input mesh (see Figure 2J). It follows

from the previous observation that, on average, the resulting polygons have at most six sides. In some cases, due to the approximate nature of the chart decomposition processes, certain charts that should meet at a single point make contact along a curve. Such typically small curves introduce small

5   segments in the coarse polygonal mesh that are unnecessary. Removing such segments is an optional step (530) that in some cases may help clean up the resulting polygonal mesh transforming it into one with faces with fewer sides (see also Figure 2K and note, for example, the removal of the short edges at the tip of the ears). Mesh cleanup may also include some form of

10   mesh optimization to better distribute the elements of the coarse mesh with respect to the input mesh. A quadrilateral mesh is then obtained from the polygonal mesh by quadrangulating it (540). In principle, any quadrangulation process may be applied, for example by simply inserting a point at the center of each face and in the middle of all edges and splitting

15   each polygon onto quads sharing a vertex at the center of that polygon. However, methods that avoid introducing a large number of extraordinary vertices (especially with high valence) are preferred. As an example, we mention the possibility of splitting every polygon with even number 2k of sides into k-1 quadrilaterals and every polygon with 2k+1 sides into k-1

20   quadrilaterals and one triangle and performing a global split after that (only

if there are any faces left that are not quadrilaterals). This has the advantage

of reducing the number of extraordinary vertices (i.e., vertices of valence

different than four). An example of a quadrangulated mesh is shown in

Figure 2L.

5

Figure 6 is a flow chart of the resampling and multiresolution analysis

process 600.

Having established a coarse quadrilateral mesh MQ, a multiresolution

10   quadrilateral subdivision surface MQSS can be extracted by resampling the

original geometry M over the faces of the coarse polygonal mesh at dyadic

positions corresponding to the finest subdivision level desired, followed by

multiresolution analysis to propagate the data from the finest level of the

subdivision hierarchy to coarser levels. In a preferred implementation this

15   could be done by first applying L-1 steps of subdivision to the quadrilateral

mesh extracted using process 500, where L is the desired number of levels in

the final multiresolution hierarchy, thus generating a fine mesh D which

constitutes a resampling domain for the final representation (610). A

resampling of the original geometry at positions corresponding to the

20   vertices of the subdivided domain mesh D yields a resampled version R of

the original mesh with subdivision connectivity (620). Multiresolution

analysis can then be used to decompose R over the faces of the base mesh,

thus generating a multiresolution quadrilateral-based subdivision surface

(MQSS) representation for the input mesh (630). For an example of

5   multiresolution analysis of meshes see H. Biermann et al., Cut-and-Paste

Editing of Multiresolution Surfaces, ACM Transactions on Graphics, vol

21(3), pp. 312-321, Proceedings of ACM Siggraph 2002, San Antonio, July

2002. An optional optimization of the resulting MQSS may also follow

(640). Figures 2M, 2N, and 2O illustrate three consecutive levels of a

10   multiresolution subdivision hierarchy for the mesh in 2A after completion of

the conversion process.

Figure 7 further illustrates the main steps of the preferred seed selection

process described in Figure 4A: Figure 7 (A) shows an approximately flat

15   region of an input model; Figure 7(B) shows samples computed in an

approximate medial axis; Figure 7 (C) illustrates the final seeds, after

discarding unnecessary samples. Region boundary samples are shown as

filled circles. Approximate medial axis samples are shown as 'x' marks.

All references cited above are herein incorporated by reference in their

entirety.

5